Welcome to The Logic Design Lab!

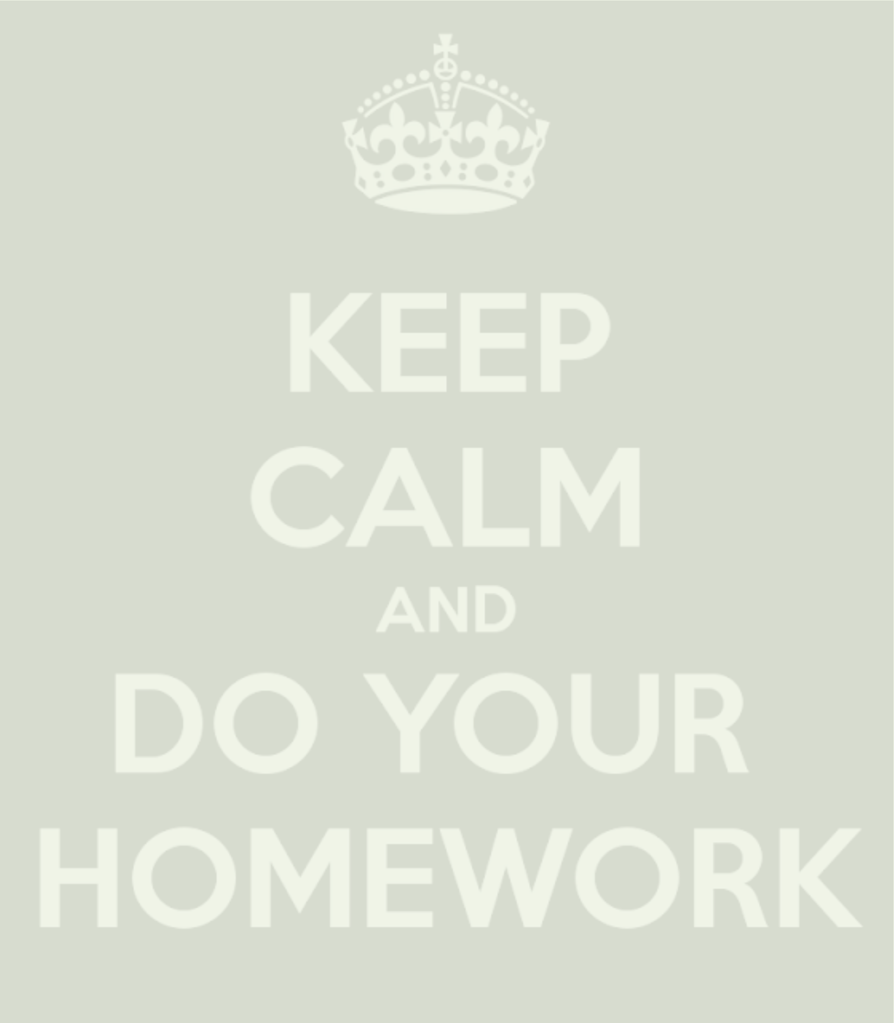
Fall 2019
Lab 5: Keyboard and Audio Modules

Prof. Chun-Yi Lee

Department of Computer Science
National Tsing Hua University

Agenda

- Lab 5 Outline
- Lab 5 Basic Questions
- Lab 5 Advanced Questions



Lab 5 Outline

- Basic questions (2%)
  - Individual assignment
  - Due on 11/14/2019. Demonstration on your FPGA board (In class)
  - Only demonstration is necessary. Nothing to submit.

- Advanced questions (5%)
  - Group assignment
  - ILMS submission due on 11/21/2019. 23:59:59.
  - Demonstration on your FPGA board (In class)
  - Assignment submission (Submit to ILMS)
    - Source codes and testbenches
    - Lab report in PDF

Lab 5 Rules

- You can use **ANY** modeling techniques

- If not specifically mentioned, we assume the following SPEC
  - **CLK** is **positive edge triggered**
  - Synchronously reset the Flip-Flops when **RESET == 1’b0**

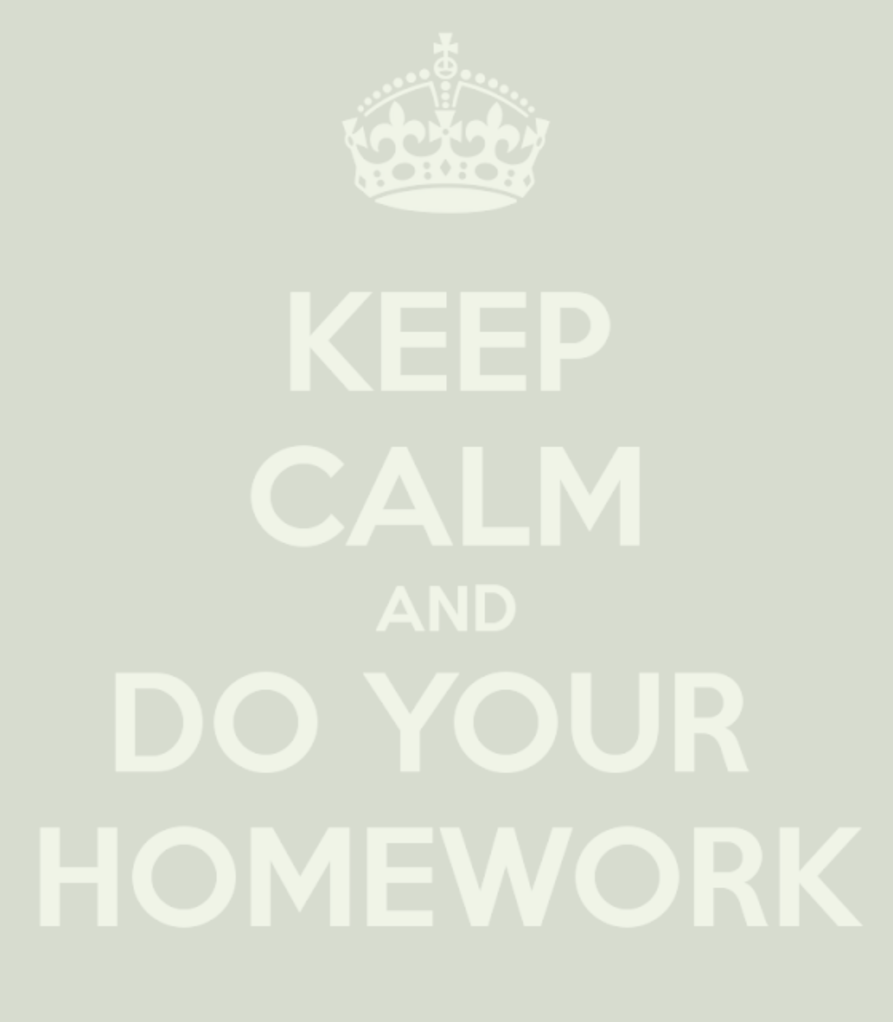
Lab 5 Submission Requirements

- Source codes and testbenches
  - Please follow the templates EXACTLY
  - We will test your codes by TAs' testbenches

- Lab 5 report
  - Please submit your report in a single PDF file
  - Please draw the **block diagrams** and **state transition diagrams** of your designs
  - Please explain your designs in detail
  - Please list the contributions of each team member clearly
  - Please explain how you test your design
  - What you have learned from Lab 5

Agenda

- Lab 5 Outline
- Lab 5 Basic Questions
- Lab 5 Advanced Questions



Basic Questions

- Individual assignment

- FPGA demonstration (due on 11/14/2019. In class.)
  - Keyboard sample code
  - Audio sample code 1 & 2
  - Mixed keyboard and audio modules together

- Demonstrate your work by waveforms

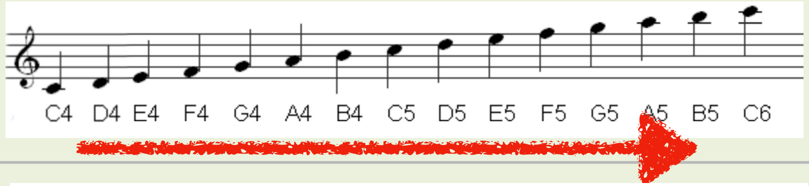
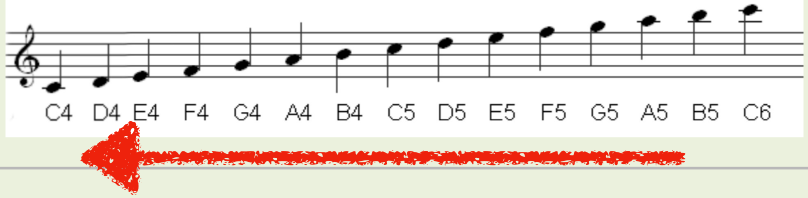
Basic FPGA Demonstration 1

- **Keyboard sample code**
  - Please implement the keyboard sample codes released on ILMS

- **Audio sample codes**
  - Please implement the audio sample codes 1 & 2 released on ILMS

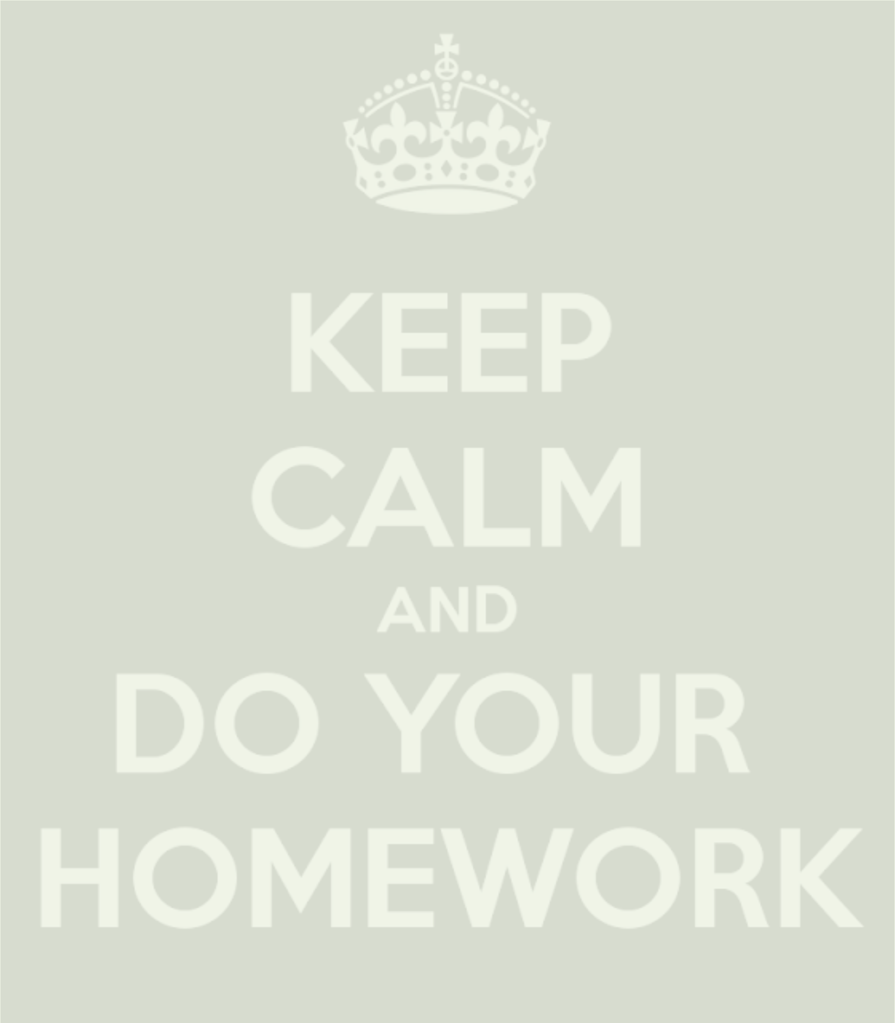
Basic FPGA Demonstration 2

- Use the numbers ("**0**" and "**1**") on the keyboard to control the scale to ascend or descend, ranging from **C4** to high **C8**.
- Change a note every **0.5 second**. If "**2**" is pressed, change a note every 1 second. If "**2**" is pressed again, go back to **0.5 second** per note.
- When it reaches **C4** or **C8**, stay on the note until the direction changes (keyboard pressed).

Button	**Direction** **Reset: Set back to C4 and ascend (0.5sec/note)** **(Use *Enter* as Reset)**
0	
1	
2	0.5 sec per note or 1 sec per note

Agenda

- Lab 5 Outline
- Lab 5 Basic Questions
- Lab 5 Advanced Questions



Advanced Questions

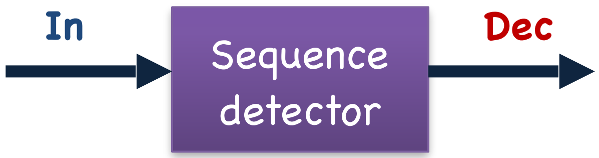
- Group assignment

- Verilog questions (due on 11/21/2019. 23:59:59.)
  - Mealy machine multi-sequence detector
  - Sliding window sequence detector
  - Traffic light controller

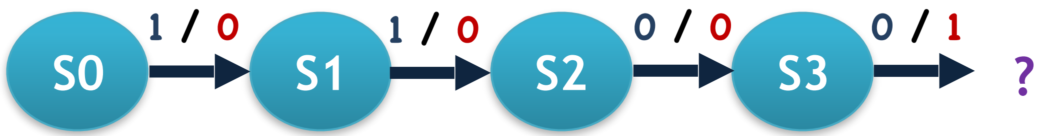
- FPGA demonstration (due on 11/21/2019. In class.)
  - Vending machine

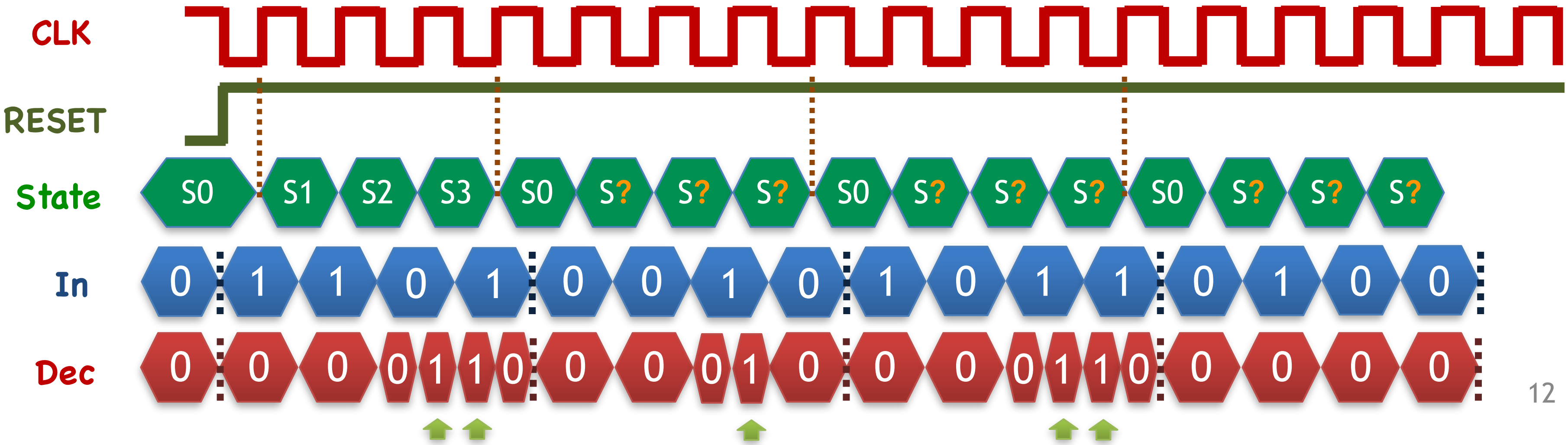
Verilog Question 1

- **Mealy machine multi-sequence detector**
  - 1-bit input **In** and 1-bit output **Dec**
  - When the four bit sequence is either **1101** or **1011**, **Dec** is set to **1**
  - Re-detect the sequence **every four bits**
  - Please draw your state diagram in your report



Complete the state diagram in your report

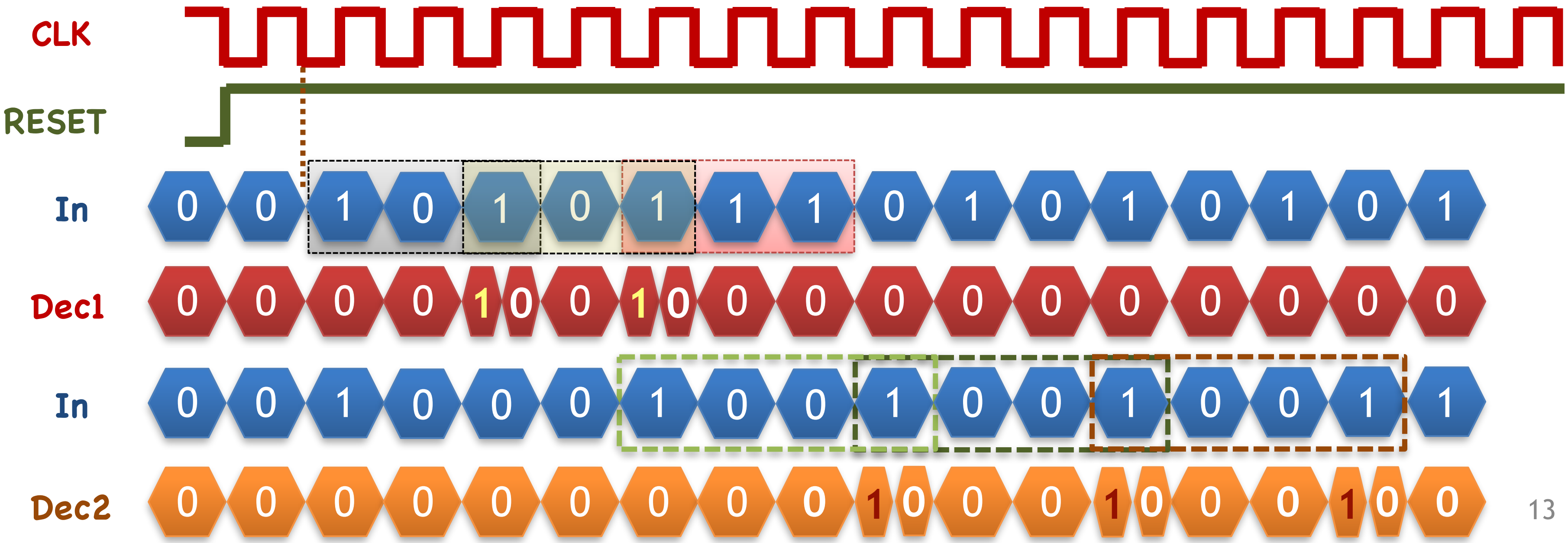




Verilog Question 2

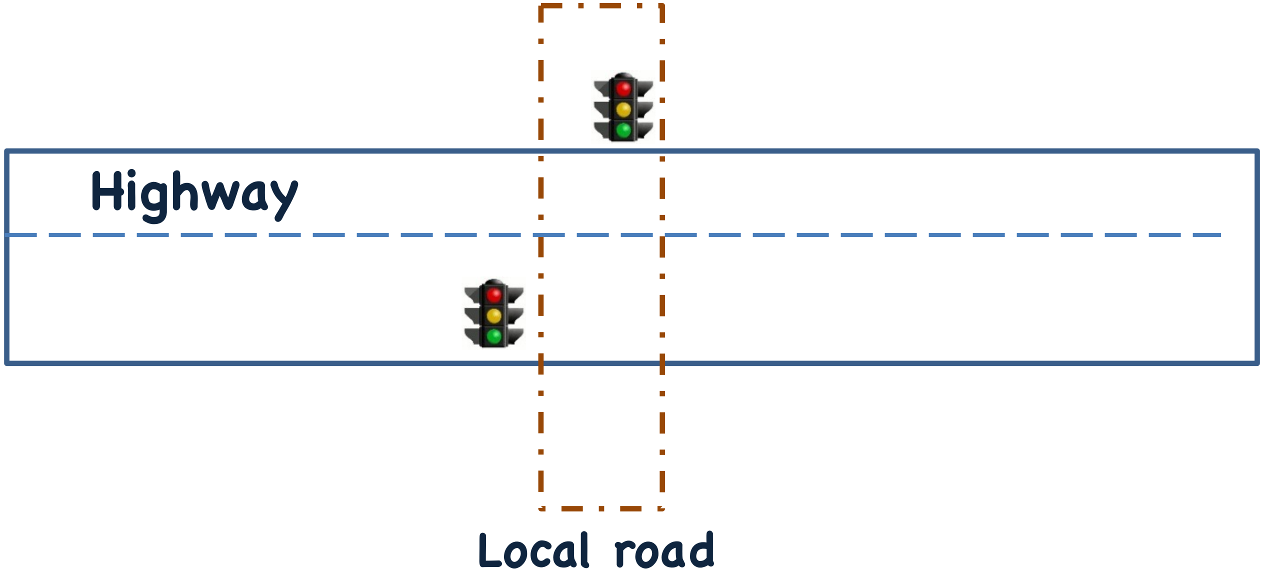
- **Sliding Window** sequence detector (**mealy machine**)
  - **Dec1** == 1’b1 when input is **101 AND no 111** occurs before
  - **Dec2** == 1’b1 when input sequence is **1001**

- **Continuous detection**
  - Detect the sequences whenever they occur
  - Please draw a state transition diagram in your report



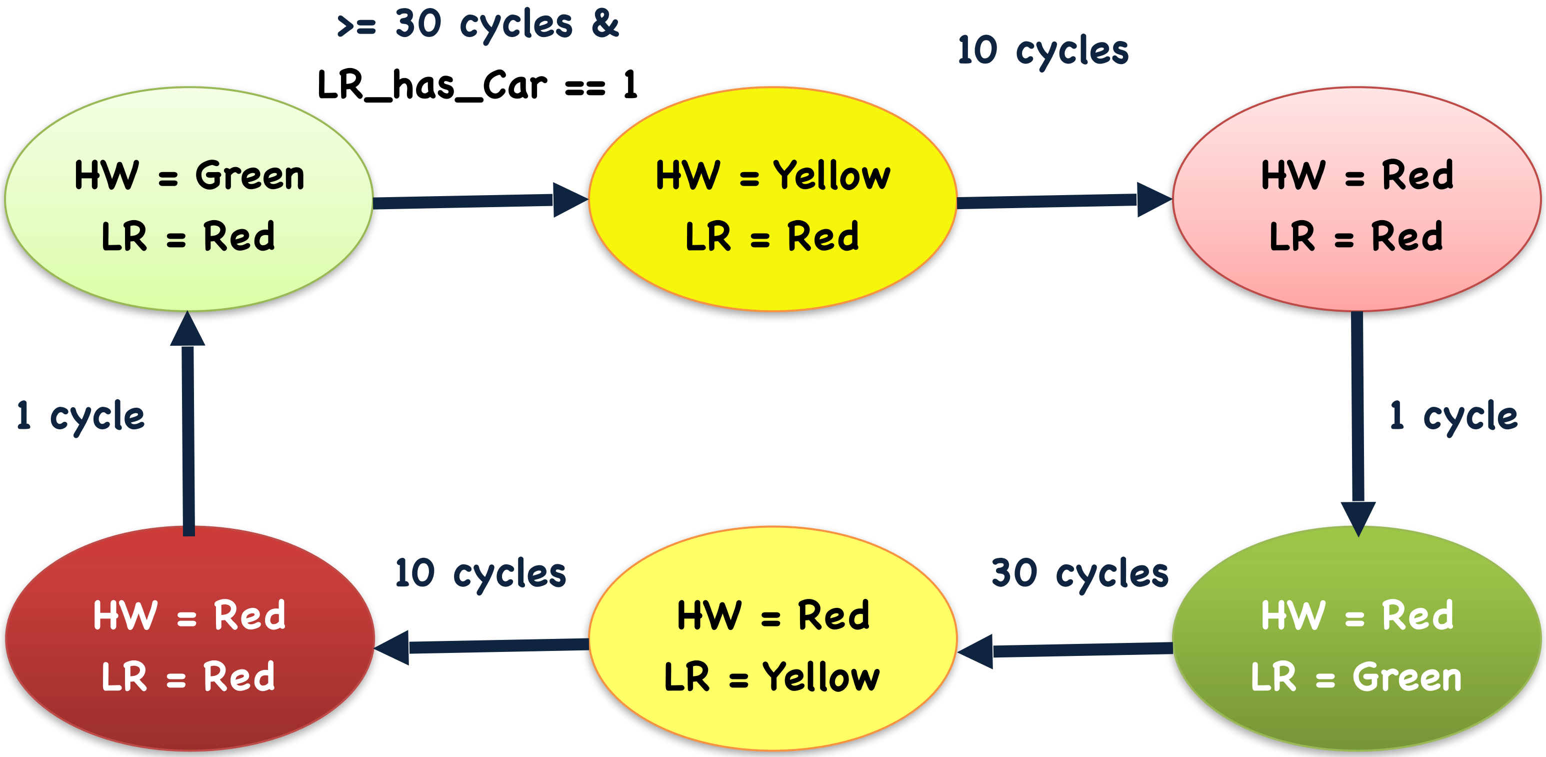
Verilog Question 3

- **Traffic light controller** for highway (**HW**) and local road (**LR**) intersection
- **HW** has higher priority and should be green as long as possible
- **LR** has a sensor to detect cars on it. When a car is sensed, LR turns green shortly
- Green light is **at least 30** clock cycles and yellow light is **10** clock cycles
- **Input**: CLK, RESET, LR_has_Car; **Output**: HW_light[2:0], LR_light[2:0]
  - HW_light & LR_light: bits [2:0] represent **Red**, **Yellow**, and **Green**, respectively

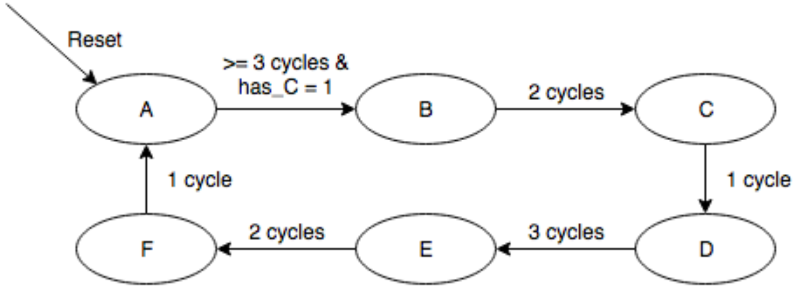


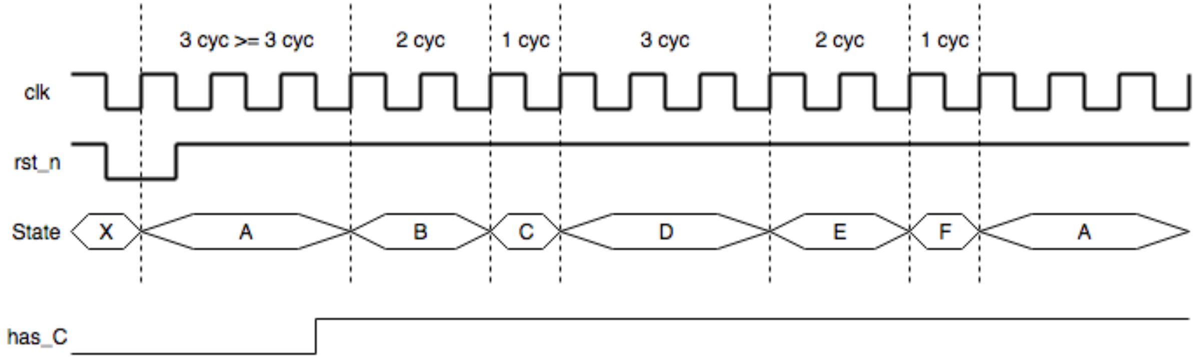
Verilog Question 3

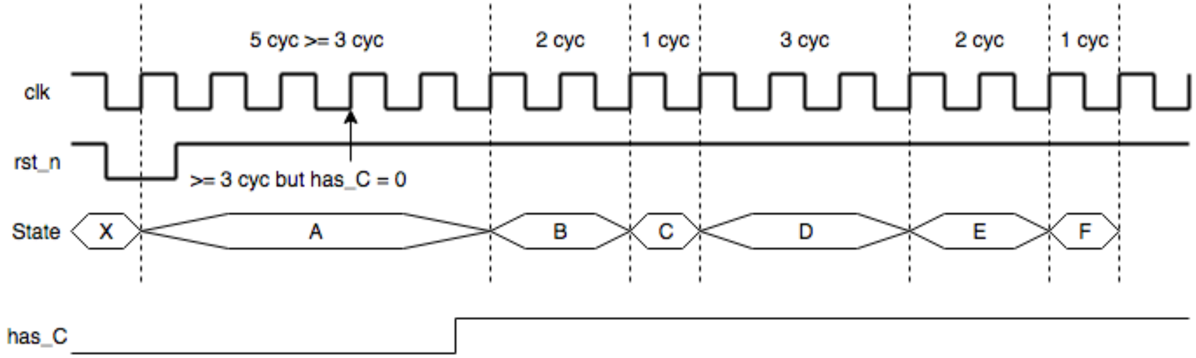
- **Traffic light controller** Finite State Machine
- **Please complete the FSM in your report** (some arrows are removed intentionally)



Verilog Question 3







- **Traffic light controller “example”** timing diagram is illustrated on the left
- **Please make sure that your state transitions follows the timing digram correctly**

Advanced Questions

- Group assignment

- Verilog questions (due on 11/21/2019. 23:59:59.)
  - Mealy machine multi-sequence detector
  - Sliding window sequence detector
  - Traffic light controller

- FPGA demonstration (due on 11/21/2019. In class.)
  - Stopwatch

FPGA Demonstration 1

- Four options available: **Coffee**, **Coke**, **Oolong**, and **Water**
  - Prices are: **Coffee** (**NT$ 60**), **Coke** (**NT$ 25**), **Oolong** (**NT$ 20**), **Water** (**NT$ 15**)

- The **rightmost two 7-segment displays** show the money inserted into the machine
  - When RESET == **1'b1**, please display "00"
  - The maximum value is **NT$ 80**

- Use **five buttons** to implement your design:
  - **Left: NT$ 5**
  - **Center: NT$ 10**
  - **Right: NT$ 50**
  - **Top: RESET**
  - **Bottom: Cancel**

FPGA Demonstration 1

- Use **four LEDs** to indicate which drinks you can buy
  - **LED[3:0]** corresponds to Coffee, Coke, Oolong, and Water, respectively
- Use the **keyboard** to select which drinks you can buy
  - **'a'**, **'s'**, **'d'**, **'f'** corresponds to **Coffee**, **Coke**, **Oolong**, and **Water**, respectively
  - Assume that the machine allows you to buy **ONLY ONE DRINK** at a time
- Use the **rightmost two 7-segment display** to **show the rest of the money** after buying a drink
  - E.g., if you inserted **NT$ 40** and bought a can of Coke (**NT$ 25**), the 7-segment display will show **NT$ 15**

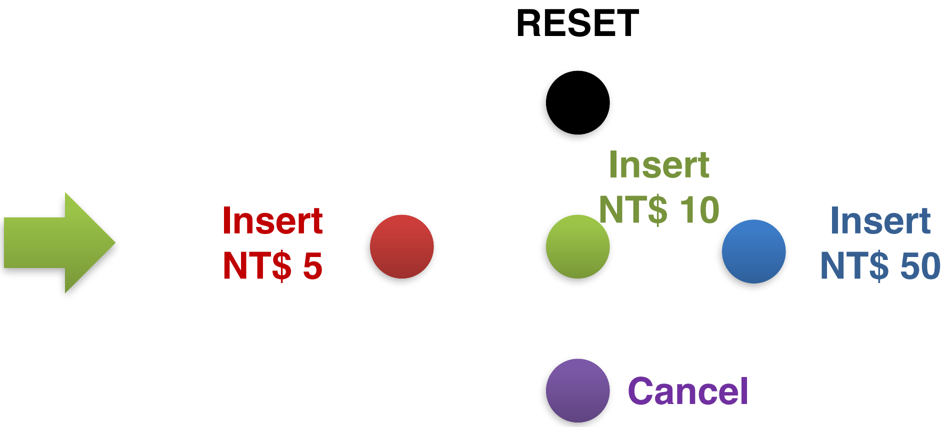
FPGA Demonstration 1

- Remember to add debounce and one-pulse circuits to your buttons

- Decrement the **7-segment display** by **NT$ 5** every second to mimic the process of returning changes
  - Return the changes until it becomes zero

- If the buyer does not want to buy a drink, he/she can use a **Cancel Button** to cancel it
  - The inserted money will be returned the same way (**NT$ 5** per second)

The layout of the buttons used in this question



Thank you for your attention!

*Lake Helen at Lassen Volcanic National Park, Shasta County, California, USA
This picture is taken by Chun-Yi Lee himself, who is also a fan of photography